A Hybrid Feature Selection Approach-Based Android Malware Detection Framework Using Machine Learning Techniques

Santosh K. Smmarwar, Govind P. Gupta, and Sanjay Kumar

Abstract With more popularity and advancement in Internet-based services, the use of the Android smartphone has been increasing very rapidly. The tremendous popularity of using the Android operating system has attracted malware attacks on these devices. Detecting variants of malware features that change their behavior to hide from being detected by the traditional method of machine learning is being an incapable and challenging task. To overcome these issues of malware feature detection, an efficient feature selection plays a crucial role in detecting malware features and reduces the dimensionality of a huge dataset and removes the unnecessary features that are not useful and keeps those relevant features that improve the classification accuracy and detection rate. To address the above issues, this paper proposed a novel framework in which a hybrid feature selection using wrapping feature selection (WFS) with the combination of random forest and greedy stepwise (RF-GreedySW) framework is devised to optimize the malware features. The proposed framework is capable of reducing a large number of attributes into an optimal feature to enhance the performance of the machine learning model. The framework used the three most popular ML classifiers such as random forest (RF), decision tree (C5.0), and support vector machine radial basis function (SVM RBF). The performance of the proposed framework is evaluated using the CIC-InvesAndMal2019 dataset. The DT (C5.0), RF, and SVM RBF model achieves better accuracy of 91.80%, 91.32%, and 82.33% on static layer, respectively. Similarly, the accuracy is 72.41%, 75.10%, and 62.07% on the dynamic layer by DT (C5.0), RF, and SVM RBF, respectively. Our model highlights good results on the CIC-InvesAndMal2019 dataset in terms of classification accuracy and increases the robustness of the model.

Keywords Machine learning · Random forest · Wrapper feature selection · Android malware detection · Ransomware · Adware · API calls

S. K. Smmarwar (✉) · G. P. Gupta · S. Kumar
Department of Information Technology, National Institute of Technology Raipur, Raipur, India
e-mail: santoshsmmarwar@gmail.com

S. Kumar
e-mail: skumar.it@nitrr.ac.in

D. P. Agrawal et al. (eds.), *Cyber Security, Privacy and Networking*, Lecture Notes in Networks and Systems 370, https://doi.org/10.1007/978-981-16-8664-1_30

1 Introduction

The rapid growth in the commercialization of Android platforms, digital services, the huge number of online service availability, and connectivity in smart devices have raised cyber-threat to user's privacy and security. These arises security concerns to the device's data privacy, integrity, and confidentiality. The attacker compromises the loopholes by installing malicious programs, uses them to access the sensitive information from the user's system. In recent times, there are more than 5 billion mobile customers as well as around 12 billion Internet of things devices are being used [1]. The increasing number of online services has attracted the threat of malware attacks. Malware is a software code having bad intension regarding the system resources, data collection, modification of codes, disguise users from normal activities for financial benefits, etc. Malware does unauthorized activities to steal valuable information, slows down the system process, consumes device memory, and sometimes demands money. There are various kinds of malware classes such as viruses, worms, Trojans, adware, spyware, Ransomware, SMSware, and many more exist [2]. Malware attacker uses evasion techniques by making the new variants of malware class to bypass the detection by using the obfuscation techniques. Two common methods used in malware analysis that is the static analysis and dynamic analysis. In static analysis, the malware is detected without running the codes. However, the static analysis is not effective to detect mutant malware [3, 4]. Some of the previous studies show that static approaches are weak in detecting new variants of malware. Instead of using static approaches, the dynamic method is capable at some level to detect the obfuscated file having a malicious nature in the virtual environment [5, 6]. However, these existing studies used the approaches like machine learning and deep learning shows some limitations like lower detection rate of malware and their category, classification accuracy, selecting the most suitable feature to predict malware [7–10]. So, in this work, we have proposed the wrapping feature selection (WFS) framework for selecting optimal features by using random forest and the greedy stepwise (RF-GreedySW) search method. The following are the main contribution of this research works as follows.

1. Proposed a novel malware detection framework in which a novel hybrid feature selection approach by combining the basic wrapping method with random forest and greedy stepwise (RF-GreedySW) search method is devised to optimize the malware features.
2. For detection of the malware, three ML classifiers such as random forest (RF), decision tree (C5.0), and support vector machine radial basis function (SVM RBF) are used.
3. Performance evaluation of the proposed framework is evaluated using the CIC-InvesAndMal2019 dataset in terms of accuracy and detection rate.

The remaining part of this paper consists of the following sections below, Sect. 2 is the related work of Android malware detection, Sect. 3 is the proposed framework,

Sect. 4 is the analysis and discussion of the results, and finally, Sect. 5 is the conclusion of the work.

2 Related Work

This section presents work related to Android malware detection approaches used in the previous studies. In an Android operating system, malware detection has done mainly based on three features like permission, intents, and API calls. The effectiveness of a malware detection system depends on the important attributes to detect efficiently variants of malware. In [11], the author worked on the detection rate of Ransomware by using a machine learning classifier from the Android-based dataset CICAndMal2017 of ten Ransomware families. The CICAndMal2017 dataset contains benign and malware applications [12] and consists of four types of malware categories as Adware, Ransomware, Scareware, and SMS Malware. In paper [13], the CICAndMal2017 dataset related to a single PCAP file was used for each malware family randomly. Similarly, in [14], authors have developed the lightweight detection system for the static feature by using the latent semantic indexing approach provides a reduced set of features to improve the detection rate. This lightweight detection system is evaluated on a machine learning classifier in which a random forest classifier is well performed. However, this work is done only for the static feature that limits the performance of the model.

3 Hybrid Feature Selection Approach-Based Android Malware Detection Framework

Here, we have proposed the hybrid feature selection approach-based Android malware detection framework. This framework used the wrapping feature selection (WFS) approach using the random forest and greedy stepwise (RF-GreedySW) search method to optimize the malware features. The dataset of CIC-InvesAndMal2019 contains the static feature and dynamic feature of malware. The static layer includes permission and intents feature, while the dynamic layer feature consists of API calls and other log files. Static layer samples contain the benign application data, and a malware category sample includes adware, premium SMS, Ransomware, scareware, and SMS malware. The dynamic layer contains malware samples such as Ransomware, scareware, SMS malware, and Adware. Figure 1 shows that the proposed wrapper feature selection framework consists of preprocessing phase, model training, and finally, the malware classification phase for malware detection, and a brief explanation of each phase is given below.

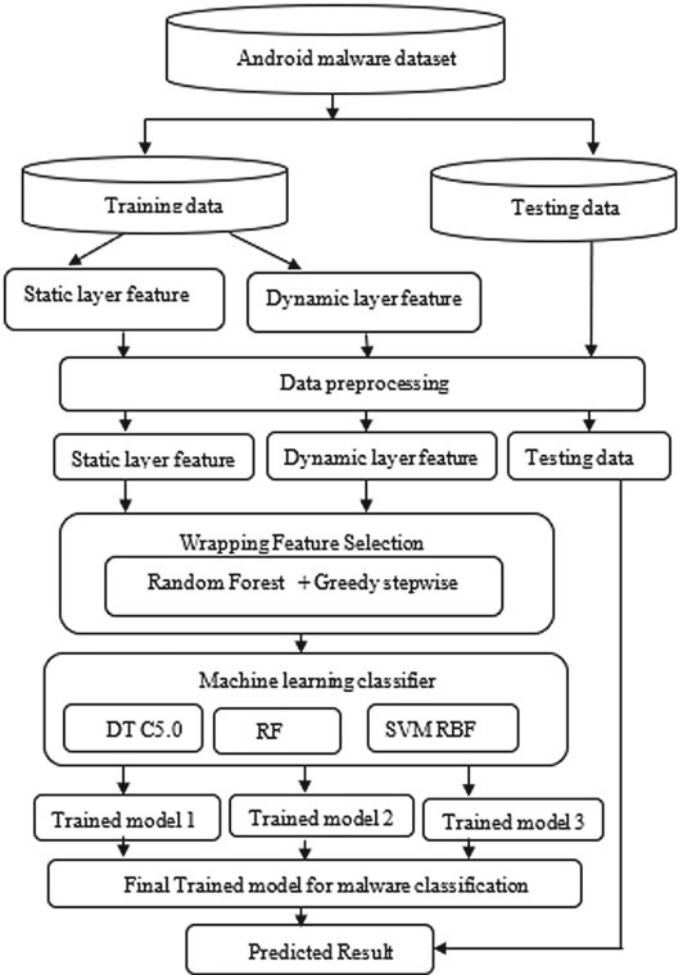

Fig. 1 Proposed framework for Android malware classification

3.1 Data Preprocessing

Preprocessing of data is the essential step to make data in a standard form for machine learning models to work well in classification. Original data is transformed into a required format, removes the missing values, and changes header name to prevent the misleading of the result. Therefore, it is necessary to transform data before going to data analysis. In our work, we removed the missing data, renaming of header name.

3.2 Wrapping Approach

The wrapping technique is used to select the best subset of features from the large number of features set using the machine learning algorithm. The wrapping approach utilized the search strategy to find a subset of features from the space vector of the feature set, and these check each selected subset based on the performance of the algorithm. The learning algorithm selects the subset of features in such a way that the obtained features are smaller than an original feature, thus provided better performance capability to the model and gives good predictive accuracy. In wrapping, we used the random forest for subset evaluator and greedy stepwise work in both directions forward or backward to get the optimal subset.

1. Random Forest: Random forest is an attribute evaluator and selects a subset of attributes sets using learning schemes. The cross-validation used to estimate the accuracy of the learning scheme for a set of attributes.
2. Greedy Stepwise: The greedy stepwise is an attribute selection algorithm and works as a greedy forward or backward search through the space of attribute subsets. It starts with selecting no/all attributes or from an arbitrary point in the space and stops working when the addition or deletion of any remaining attributes results in a decrease in evaluation. This can also produce a ranked list of attributes by traversing the space from one side to the other and recording the order that attributes are selected.

3.3 Machine Learning Algorithm

This section discussed some of the basic machine learning classifiers that were employed on the Android dataset to measure the performance of our approach as well as accuracy.

(a) **Decision Tree (C5.0)**: This is the classification model of supervised learning used to create a binary tree or multi-branches tree. It was developed in the year 1994 by Ross Quinlan used the information gain or entropy for data splitting. C5.0 is used to solve various kinds of problems by using the automatic learning process to tackle the numeric, nominal, and missing values, provide the best result by partitioning the dataset into small subparts. It is useful for high-dimensional datasets to predict relevant and irrelevant features for classification purposes.
(b) **Random Forest (RF)**: Random forest algorithm is the most efficient supervised learning classifier to predict the accurate result. It generates multiple decision trees by using bootstrap samples in resampling training data and follows the ensemble learning approach to handle the complex and difficult problems for improving the prediction accuracy of the model. The ensemble learning approach combines the weak learner into the strong learner.

(c) **Support Vector Machine (SVM RBF)**: SVM is a state-of-the-art classification model, used the RBF as a computational high power kernel-based tool for classification. It is used in various areas due to its high accuracy capability and handles high-dimensional data. SVM aims to maximize the hyperplane so that more features are separated. The kernel function used hyperparameters known as gamma and regularization parameters. The gamma values are used to improve the accuracy of the model, and the regularization value reduces the misclassification of data points.

4 Result Analysis and Discussion

The performance evaluation of our proposed framework is done on the CIC-InvesAndMal2019 Android dataset. The work is classified into two parts for the classification of Android malware that is on a static layer and dynamic layer.

4.1 Experimental Setup and Evaluation Parameter

In this work, the proposed framework used the Java-based environment Weka 3.8.4 tool for feature selection and optimization. The experiment was performed on Windows 10 with a configuration of Intel core i3-2330 processor 2.20 GHz with 8 GB RAM and using the R tool. The performance parameter and experimental setup have the main role to analyze the effectiveness of the machine learning model. We have taken datasets for training and testing in the ratio of 80:20, respectively, and calculated the accuracy, sensitivity, specificity, kappa statistics, and AUC-ROC values for evaluation of our framework as mentioned in [15, 16].

4.2 Static Layer Malware Category Detection

Table 1 shows the accuracy and kappa statistics of different machine learning classifiers evaluated on the CIC-InvesAndMal 2019 dataset. The accuracy obtained by all three classifiers DT, RF, and SVM RBF is 91.80, 91.32, and 82.33%. Among all three classifiers, the best accuracy is obtained by the DT classifier.

The kappa statistics of the machine learning model are used to assess the classification performance of the model. The kappa statistics are computed by all three models as 79.56%, 77.52%, and 50.12% by DT, RF, and SVM, respectively, on the static layer. The AUC-ROC curve is 0.95, 0.93, and 0.90 of ML models as shown in Fig. 2 of DT, RF, and SVM, respectively, indicating the better performance of the model. This shows the significant improvement in the overall performance of the malware detection rate.

Table 1 Comparison of accuracy and kappa statistics on static layer for malware category classification

ML classifier	Accuracy (%)	Kappa statistics (%)
DT (C5.0)	**91.80**	**79.56**
RF	91.32	77.52
SVM RBF	82.33	50.12

Highest result shown in bold values

Table 2 Comparison of sensitivity and specificity on static layer

Malware category	DT (C5.0)		RF		SVM RBF	
	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
Adware	71.05	98.82	68.42	99.16	44.73	98.15
Benign	98.10	79.37	97.47	80.63	98.73	53.75
PremiumSMS	90.00	99.83	95.0	99.18	95.00	99.83
Ransomware	89.18	99.16	86.48	98.49	10.05	100
Scareware	56.08	99.32	56.09	99.15	04.87	98.65
SMS malware	66.66	99.67	58.33	99.50	66.66	97.04

Table 2 demonstrated the sensitivity and specificity of a state-of-the-art machine learning classifier with optimizing the feature of the android dataset on the static layer. The sensitivity values of malware range 56.08–98.10% for DT, 56.09–97.47% for RF, and 04.87–98.73% for SVM RBF. The specificity values of the malware class are 79.37–99.83% for DT (C5.0), 80.63–99.50% for RF, and 53.75–100% for SVM RBF.

4.3 Dynamic Layer Malware Category Detection

Table 3 demonstrated an accuracy and kappa statistics comparison of three ML models are evaluated on the CIC-InvesAndMal2019 dataset. The accuracy achieved by these models is 72.41%, 75.10%, and 62.07 by DT, RF, and SVM RBF, respectively, on tenfold cross-validation, and the highest accuracy is achieved by RF models.

The kappa statistics of ML models in Table 3 is to be computed as 62.92% is highest for DT (C5.0), 61.64% of RF, and 44.38% of SVM RBF. Figures 2 and 3 represent the ROC comparison chart of tenfold CV models for all models. The ROC curve of each model is plotted simultaneously. Area under the curve (AUC) measures the area under an entire ROC curve. If the value of AUC-ROC is found greater than 0.5, a model is considered better and appropriate for developing a prediction model. The AUC-ROC value of the three ML classifiers comes out to be 0.97 for RF, 0.99 for DT, and 0.71 for SVM RBF. The AUC-ROC value of the DT model is 0.99 which is

Table 3 Comparisons of accuracy and kappa statistics on the dynamic layer for malware category classification

ML classifier	Accuracy (%)	Kappa statistics
DT (C5.0)	72.41	62.92
RF	**75.10**	**66.26**
SVM RBF	62.07	50.32

Highest result shown in bold values

Table 4 Comparison of sensitivity and specificity on the dynamic layer

Malware category	DT (C5.0)		RF		SVM RBF	
	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
Adware	69.57	88.17	60.87	92.47	78.26	68.82
Ransomware	83.33	93.48	79.17	91.30	70.83	90.22
Scarware	59.38	91.67	65.62	84.52	18.75	92.85
SMSmalware	78.38	89.87	78.38	93.67	70.27	93.67

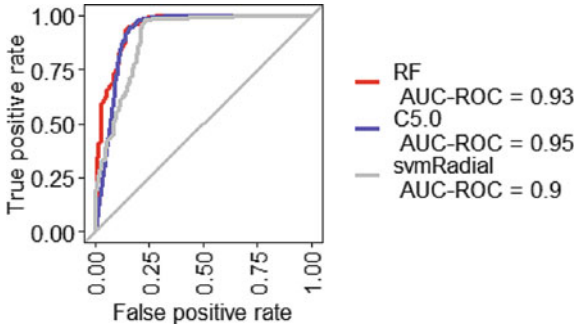

Fig. 2 ROC curve for tenfold cross-validation on static layer

far greater than 0.5 implies that the proposed model including other models is good to build a prediction model and not fall under random guesser.

The results from Table 4 contain the comparison of sensitivity and specificity values on the dynamic layer. Sensitivity values for adware, Ransomware, scareware, and SMS malware are to be computed by three machine learning models to test the performance of the model. The highest sensitivity value is 83.33%, and the lowest is 59.38% for the Ransomware malware by DT (C5.0) model as compared to other classifiers (RF, SVM RBF). The sensitivity values of the RF model for Ransomware are 78.38% which is the highest and 60.87% is the lowest. The sensitivity value of another classifier by SVM RBF of 78.26% is the highest for adware, and 18.75% is the lowest for scareware.

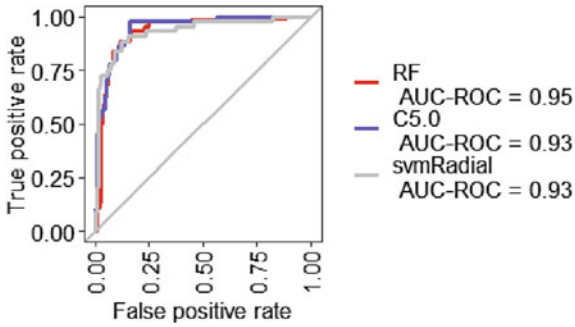

Fig. 3 ROC curve for tenfold cross-validation on the dynamic layer

5 Conclusion

This research work proposed a novel malware detection framework in which a novel hybrid feature selection approach by combining the wrapping method with random forest and greedy stepwise (RF-GreedySW) search method is devised to optimize the malware features. Our study uses the most popular machine learning models such as DT (C5.0), RF, and SVM RBF to identify malware types using the latest Android dataset known as CIC-InvesAnd2019. The potential application of our approach can be in the problems like object identification and image segmentation where feature selection is a challenging task. From the above result, we can be concluded that our proposed framework is effective and efficient in malware detection. In the future, we plan to implement our framework based on deep learning techniques using different real-time datasets.

References

1. Imtiaz SI, ur Rehman S, Javed AR, Jalil Z, Liu X, Alnumay WS (2021) DeepAMD: detection and identification of Android malware using high-efficient Deep Artificial Neural Network. Future Gener Comput Syst 115:844–856
2. Vasan D, Alazab M, Wassan S, Naeem H, Safaei B, Zheng Q (2020) IMCFN: image-based malware classification using fine-tuned convolutional neural network architecture. Comput Netw 171:107138
3. Venkatraman S, Alazab M (2018) Use of data visualization for zero-day malware detection. Secur Commun Netw
4. Shafiq M, Tian Z, Bashir AK, Du X, Guizani M (2020) IoT malicious traffic identification using wrapper-based feature selection mechanisms. Comput Secur 94:101863
5. Alzaylaee MK, Yerima SY, Sezer S (2020) DL-Droid: deep learning-based android malware detection using real devices. Comput Secur 89:101663
6. D'Angelo G, Palmieri F, Robustelli A, Castiglione A (2021) Effective classification of Android malware families through dynamic features and neural networks. Conn Sci 1–16
7. Tchakounté F, Djakene Wandala A, Tiguiane Y (2019) Detection of Android malware based on sequence alignment of permissions. Int J Comput (IJC) 35(1):26–36

8. Yuan Z, Lu Y, Xue Y (2016) DroidDetector: Android malware characterization and detection using deep learning. Tsinghua Sci Technol 21(1):114–123
9. Yuan Z, Lu Y, Wang Z, Xue Y (2014) Droid-Sec: deep learning in android malware detection. In: Proceedings of the 2014 ACM conference on SIGCOMM, Aug 2014, pp 371–372
10. Jerbi M, Dagdia ZC, Bechikh S, Said LB (2020) On the use of artificial malicious patterns for android malware detection. Comput Secur 92:101743
11. Noorbehbahani, F., Rasouli, F., & Saberi, M. (2019, August). Analysis of machine learning techniques for ransomware detection. In 2019 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC) (pp. 128-133). IEEE
12. Android malware dataset (CIC-AndMal2017) (2021) https://www.unb.ca/cic/datasets/andmal2017.html. Accessed 2021-05-11
13. Chen R, Li Y, Fang W (2019) Android malware identification based on traffic analysis. In: International conference on artificial intelligence and security, July 2019. Springer, Cham, pp 293–303
14. Singh AK, Wadhwa G, Ahuja M, Soni K, Sharma K (2020) Android malware detection using LSI-based reduced opcode feature vector. Procedia Comput Sci 173:291–298
15. Kumar P, Gupta GP, Tripathi R (2021) Toward design of an intelligent cyber attack detection system using hybrid feature reduced approach for IoT networks. Arab J Sci Eng 46(3):3749–3778
16. Kumar P, Gupta GP, Tripathi R (2021) Design of anomaly-based intrusion detection system using fog computing for IoT network. Autom Control Comput Sci 55(2):137–147